Standard Iterative Solvers of SLE

- iterative solution of large linear systems is one of the most important numerical tasks in scientific computing (they occur in the discretization of both ODE and PDE)
- direct solvers are often not competitive:
  - too large number of unknowns (cf. PDE in 3D)
  - sparse matrices (classical elimination destroys sparsity)
  - anyway only approximations, thus no need for exact solution (this holds esp. in the *nonlinear* case, where a SLE occurs in each step of some outer iteration for the nonlinearity)
- objective: „for 3 digits, you need 10 steps“ – no matter how big the number n of unknowns is
- however typically: speed of convergence deteriorates with increasing n!

Principal Remarks on Iterations

- consider an iterative scheme, starting from $x^{(0)}$, and, hopefully converging to the solution x of Ax=b:
  $$x^{(0)} \to x^{(1)} \to \ldots \to x^{(i+1)} \to \ldots \to \lim_{i\to\infty} x^{(i)} = x$$
- speed of convergence: $\left\|x - x^{(i+1)}\right\| < \gamma \cdot \left\|x - x^{(i)}\right\|$ for some $0 < \gamma < 1$
- typical behaviour of iterative schemes:
  $$\gamma = O(1 - n^{-k}),\ k \in \{0,1,2,\ldots\}$$
- strategy: look for iterative methods with
  - only O(n) arithmetic operations per step of iteration (cost)
  - a convergence behaviour like $\gamma < 1 - \text{const.}$
- two big families: relaxation and Krylov subspace methods

Relaxation Techniques 1

- sometimes also called smoothing methods:
  - Richardson iteration
  - Jacobi iteration
  - Gauß-Seidel iteration
  - successive over relaxation (SOR) or damped methods
- All start from the *residual* r after i steps of iteration:

$$r^{(i)} = b - Ax^{(i)} = Ax - Ax^{(i)} = A\left(x - x^{(i)}\right) = -Ae^{(i)}$$

(the error e is not known, so r is used as an indicator)

- How to obtain an improvement?
  - Richardson: use the residual as it is as a correction
  - Jacobi/Gauß-Seidel: make one component of r vanish
  - SOR/damped: same, but do a bit less/more than indicated

Relaxation Techniques 2

- Richardson iteration:

$$\text{repeat}(i):\ \text{for } k = 1,\ldots n \text{ do } x_k^{(i+1)} = x_k^{(i)} + r_k^{(i)}$$

- (damped) Jacobi iteration:

$$\text{repeat}(i):\ \text{for } k = 1,\ldots n \text{ do } y_k = r_k^{(i)} / a_{k,k}$$

$$\text{for } k = 1,\ldots n \text{ do } x_k^{(i+1)} = x_k^{(i)} + \alpha \cdot y_k$$

(compute and store updates, apply them at the end)

- Gauß-Seidel or SOR, resp.:

$$\text{repeat}(i):\ \text{for } k = 1,\ldots n \text{ do } x_k^{(i+1)} = x_k^{(i)} + \alpha \cdot r_k^{(i)} / a_{k,k}$$

(compute same updates, but apply them at once)

- For an analysis, decompose A in its strictly lower, diagonal, and strictly upper part: $A = L_A + D_A + U_A$

Relaxation Techniques 3

- all can be written in the form $Mx^{(i+1)} + (A-M)x^{(i)} = b$
  or $x^{(i+1)} = M^{-1}b - \left(M^{-1}A - I\right)x^{(i)} = x^{(i)} + M^{-1}r^{(i)}$
- how looks M?
  - Richardson: $M = I$
  - Jacobi: $M = D_A$
  - Gauß-Seidel: $M = D_A + L_A$
  - SOR: $M = \alpha^{-1} \cdot D_A + L_A$
- some convergence results:
  - If the iteration converges at all, then towards x.
  - Crucial quantity is the *spectral radius* of $-M^{-1} \cdot (A-M)$ which is smaller than 1 if and only if the iteration converges.
  - necessary for SOR: $0 < \alpha < 2$
  - sufficient for Gauß-Seidel/SOR: A positive definite
  - sufficient for Jacobi: both A and $2D_A - A$ are pos. def.

Towards Krylov: Steepest Descent

- alternative point of view for positive definite A:
  x solves $Ax = b \Leftrightarrow x$ minimizes $f(x) = 0.5 \cdot x^T Ax - b^T x + c$
  (uniqueness of minimum due to positive definiteness)
- hence new strategy: look for minimum of f; possible way: method of *steepest descent* (looks for the best improvement in the direction of the negative gradient)

$$\text{repeat}(i): \alpha_i = \frac{r^{(i)^T} r^{(i)}}{r^{(i)^T} A r^{(i)}}; \quad x^{(i+1)} = x^{(i)} + \alpha_i r^{(i)}; \quad r^{(i+1)} = r^{(i)} - \alpha_i A r^{(i)};$$

  (1D minimization along search direction $-f'(x^{(i)}) = r^{(i)}$)
- even simpler: search along coordinate axes (is GS!)
- slow convergence (progress may get lost again!)
- crucial quantity: spectral condition number of A

$$\kappa(A) = \lambda_{\max}(A) / \lambda_{\min}(A)$$

Improvement: Conjugate Directions

- further enhancement:
  - orthogonal search directions, error after i steps shall be orthogonal to all previous search directions
  - nothing destroyed, hence: in principle a direct method
  - however, since n iterations are too much in practice: used as an iterative method (therefore called *semi-iterative* method)
  - new search directions: $x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}$
  - optimum orthogonality: $0 = d^{(i)^T} e^{(i+1)}$, but error is missing ☹
  - therefore *conjugation*: $0 = d^{(i)^T} A e^{(i+1)}$
    (u and v are called *A-orthogonal* or *conjugate*, if $u^T A v = 0$)
  - algorithm: start with $d^{(0)} = r^{(0)}$ and iterate:

$$\text{repeat}(i): \alpha_i = \frac{d^{(i)^T} r^{(i)}}{d^{(i)^T} A d^{(i)}}; \quad x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}; \quad r^{(i+1)} = r^{(i)} - \alpha_i A d^{(i)};$$

  - still to be done: construction of the conjugate directions $d^{(i)}$

Finally: Conjugate Gradients

- above method + construction of the conjugate directions
- principle of construction: Gram-Schmidt conjugation of r's
- no detailed derivation here, just the algorithm:

$$\text{repeat}(i): \alpha_i = \frac{d^{(i)^T} r^{(i)}}{d^{(i)^T} A d^{(i)}}; \quad x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}; \quad r^{(i+1)} = r^{(i)} - \alpha_i A d^{(i)};$$

$$\beta_{i+1} = \frac{r^{(i+1)^T} r^{(i+1)}}{r^{(i)^T} r^{(i)}}; \quad d^{(i+1)} = r^{(i+1)} + \beta_{i+1} d^{(i)};$$

- faster than steepest descent, but still depending on n!
- search spaces form a so-called *Krylov sequence* :

$$\text{span}\{d^{(0)}, \ldots, d^{(i-1)}\} = \text{span}\{d^{(0)}, A d^{(0)}, \ldots, A^{i-1} d^{(0)}\} = \text{span}\{r^{(0)}, A r^{(0)}, \ldots, A^{i-1} r^{(0)}\}$$

- other famous Krylov methods: GMRES, Bi-CGSTAB